RubiX and Locality Aware Scheduling

PRESTO SUMMIT 2019, Bangalore

RubiX

Why Caching

- Popularity of Cloud Stores like S3
  - \+ Near-infinite capacity
  - \+ Inexpensive
  - \+ Ease of use
  - \- Network Latencies
  - \- Back-offs

RubiX

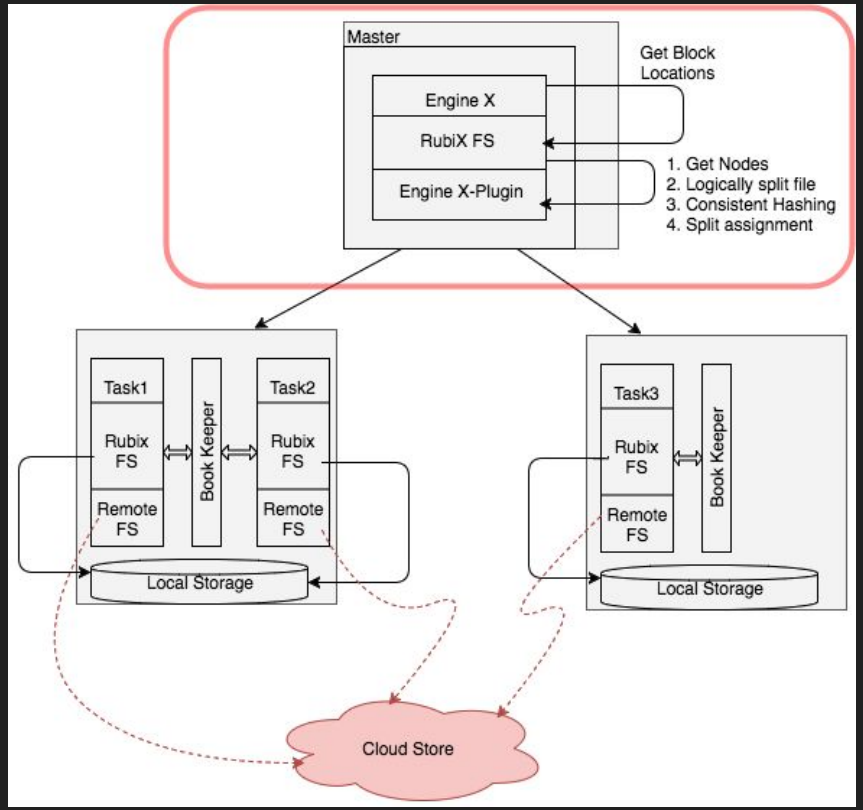
- Extendible to many engines
- Columnar format friendly
- Works well with autoscaling
- Share-able across engines/instances

Architecture

- Split ownership assignment system
- Data Caching System
- Plugins

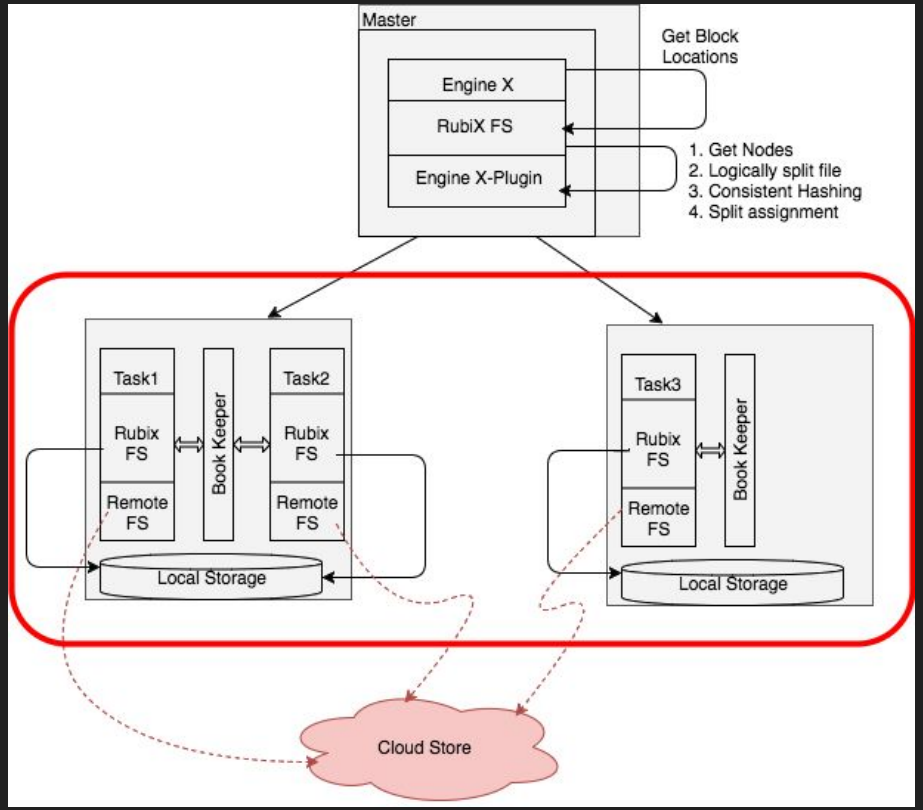
Architecture

- Split ownership assignment system
  - Used in master node during split computation
  - Calculates which node owns particular split of file
  - Uses Consistent Hashing to work well with Autoscaling



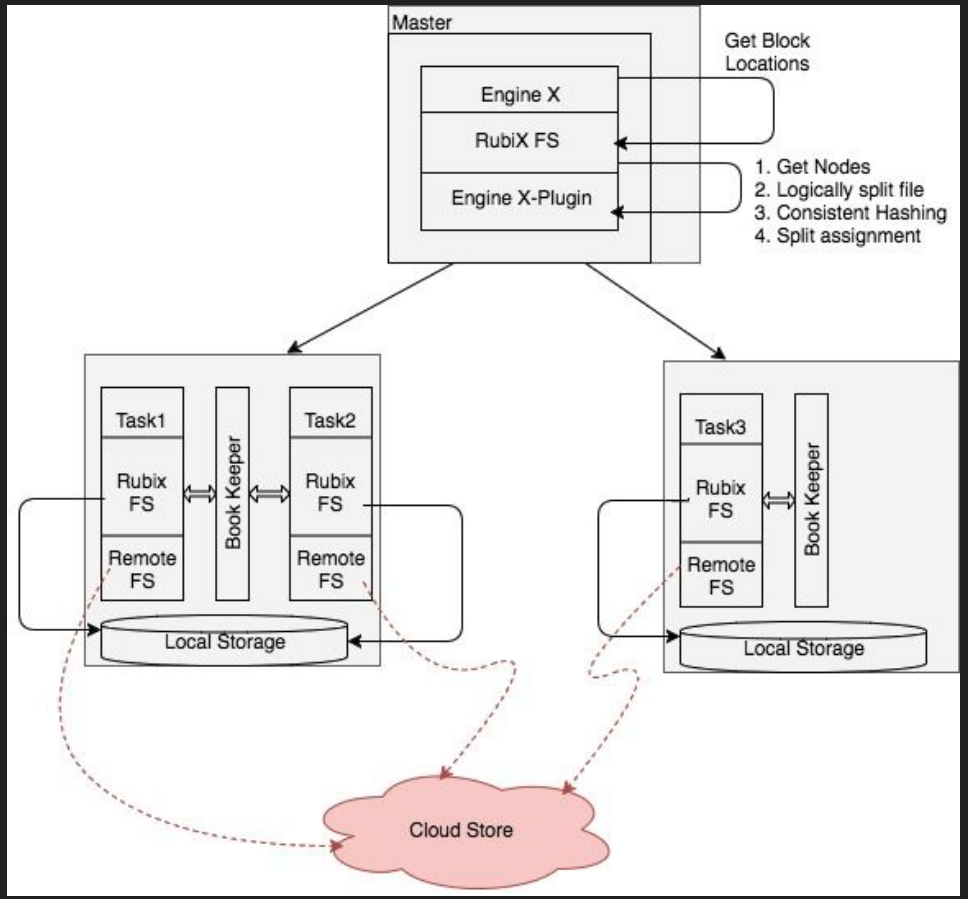
Architecture

- Data Caching System
  - Used in worker nodes when data is read
  - Read from disk or remote as per the metadata
  - Metadata stored in units of block (1MB each)
  - **BookKeeper** provides metadata for the block
  - Metadata too Checkpointed to local disk



Architecture

- Plugin
  - Provides two types of information
    - How to get the list of nodes in the system
    - FileSystem for remote reads
  - E.g. presto plugin, hadoop1 plugin, hadoop2 plugin



Locality Aware Scheduling in Presto

Presto Scheduler

- Split assignment to nodes
- Degree of parallelism
- Uniform distribution

Presto Scheduler: Advantages

- * Uniform division of workload
- Dynamic: Splits occur in batches

Scope for improvement?

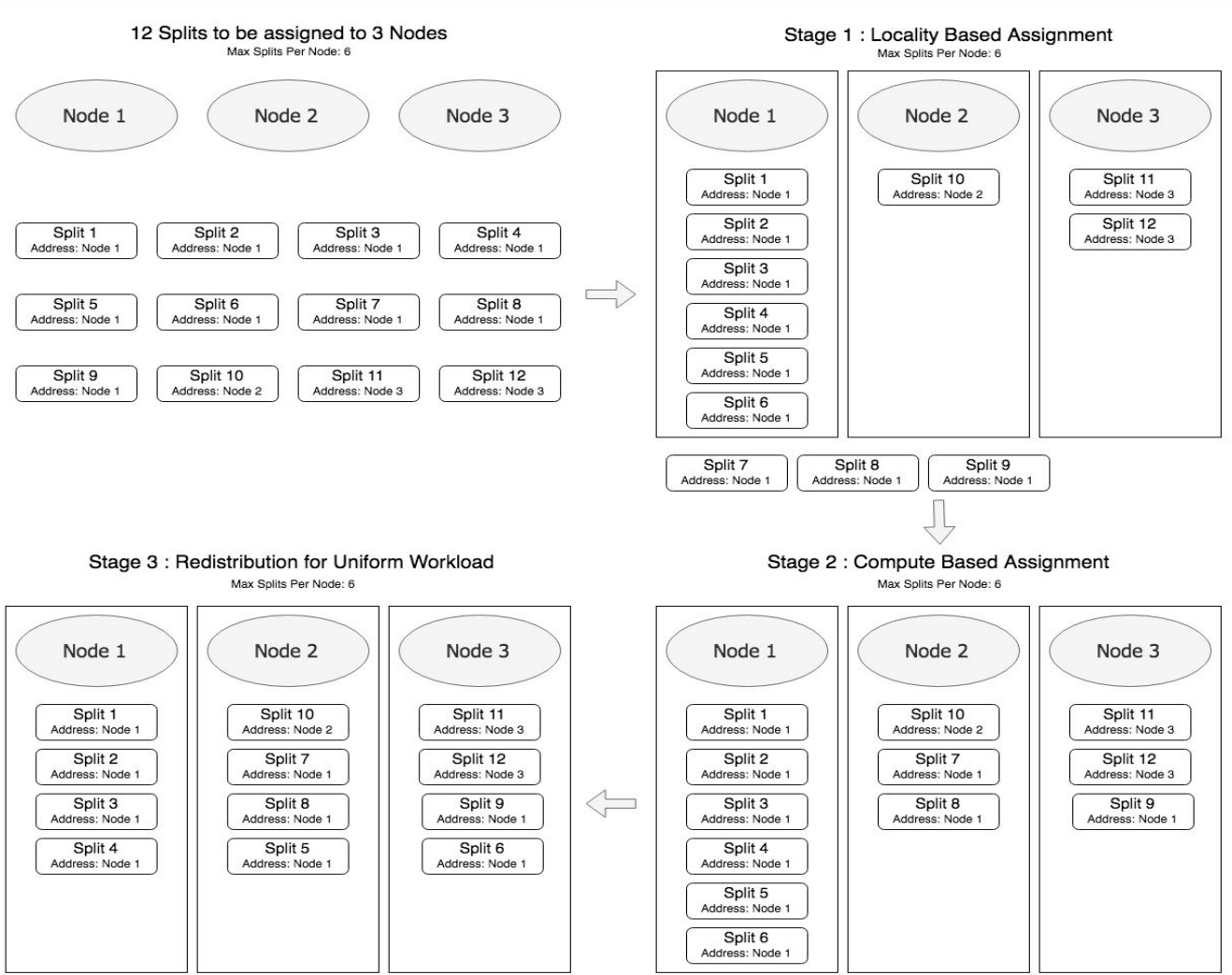
- Uniform division of workload => Uniform query execution time?
- Data locality: What if the data resides within the disk or in a cache?
- Impact on query performance and network traffic?

Locality Aware Scheduling

- Locality as an extra dimension for Split assignment
- Ensure uniform workload
- Multi-stage best effort implementation

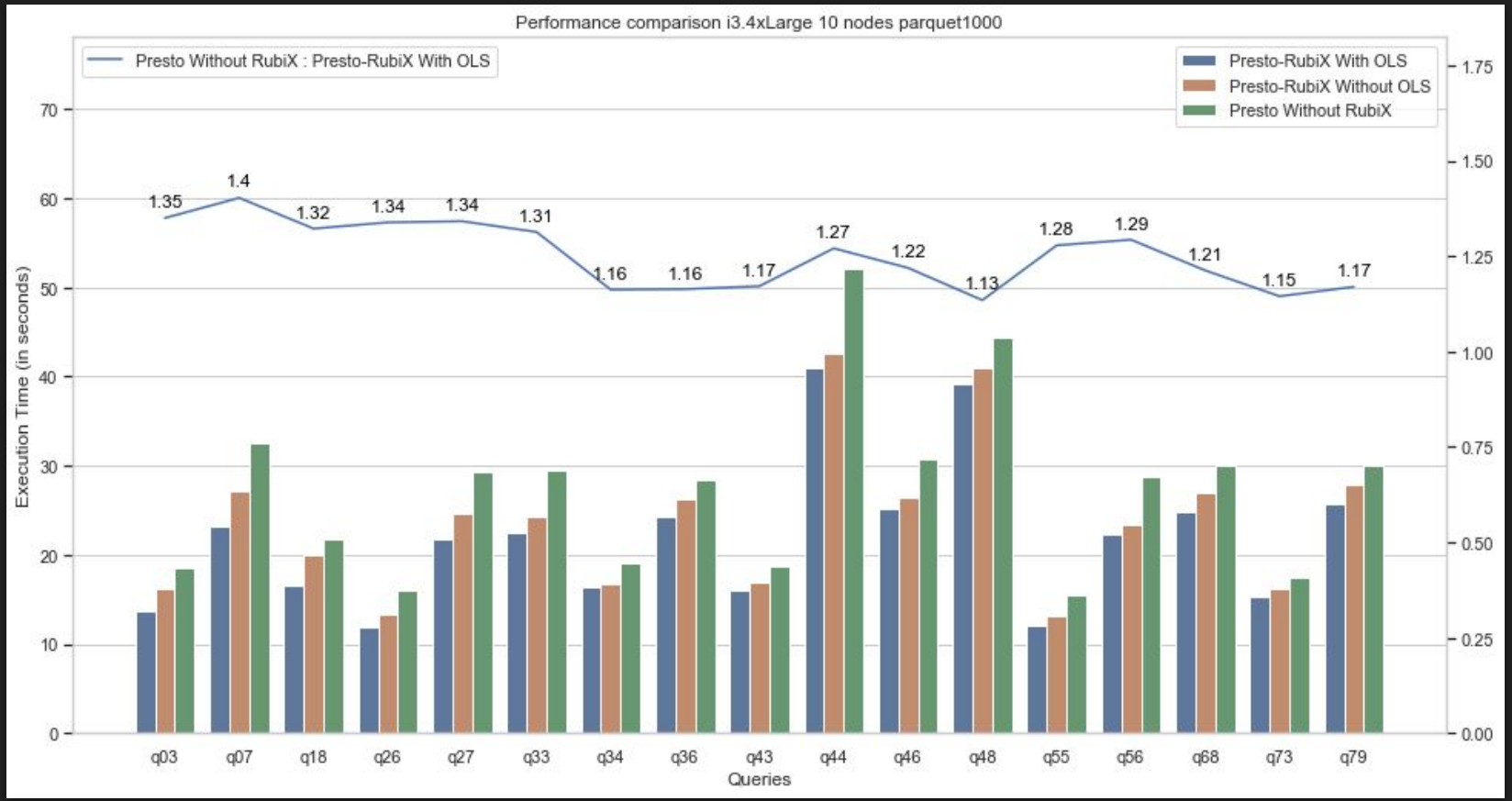
Locality Aware Scheduling: Implementation

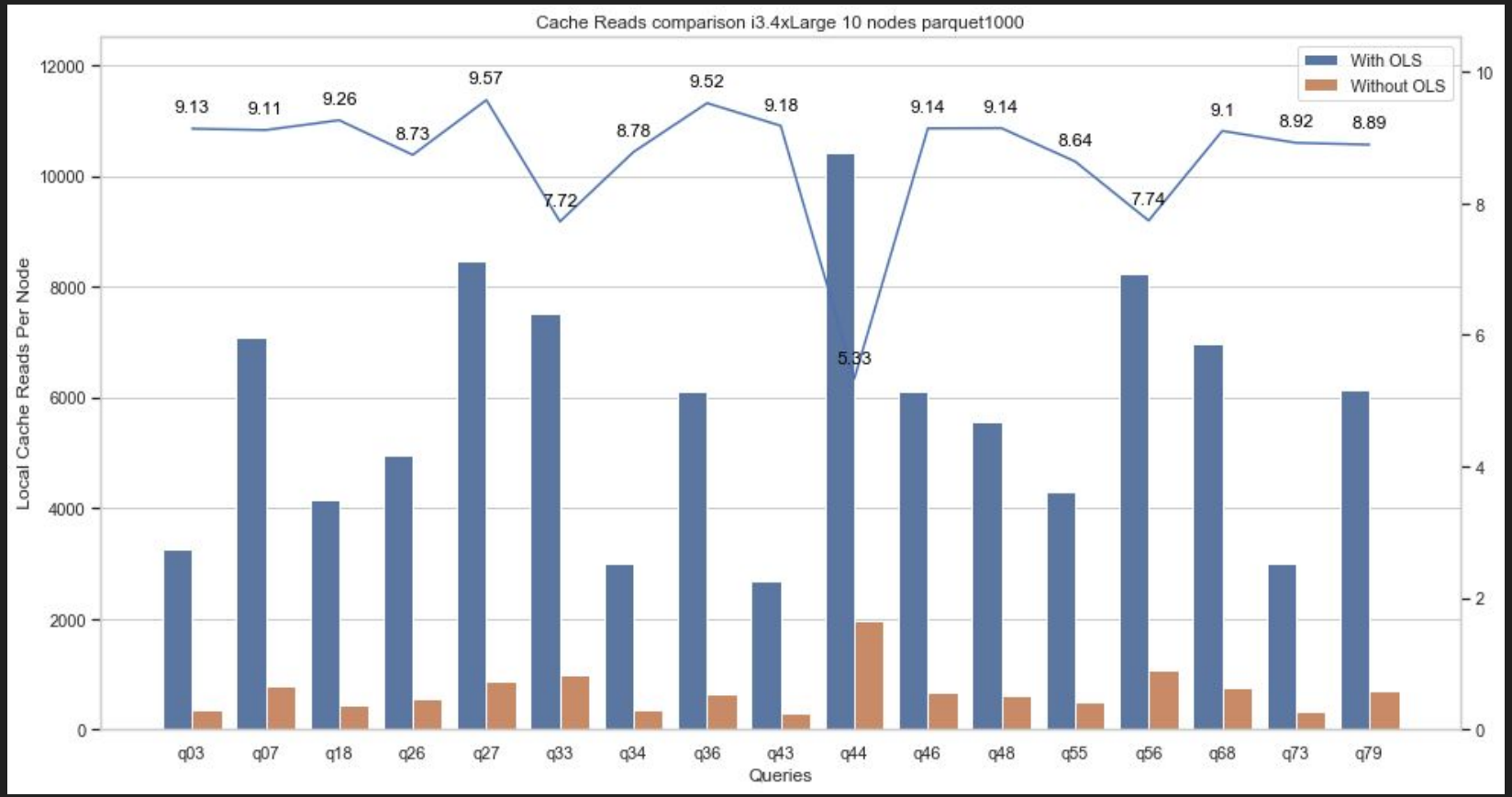
- Three Stages:
  - Stage 1: Assignment based on locality
  - Stage 2: Compute based assignment: Allocate splits to nodes with max slots available (existing behaviour)
  - Stage 3: Redistribution of splits if non-uniform assignment


12 Splits to be assigned to 3 Nodes
Max Splits Per Node: 6
Node 1
Node 2
Node 3
Split 1 Address: Node 1
Split 2 Address: Node 1
Split 3 Address: Node 1
Split 4 Address: Node 1
Split 5 Address: Node 1
Split 6 Address: Node 1
Split 7 Address: Node 1
Split 8 Address: Node 1
Split 9 Address: Node 1
Split 10 Address: Node 2
Split 11 Address: Node 3
Split 12 Address: Node 3
Stage 1 : Locality Based Assignment
Max Splits Per Node: 6
Node 1
Node 2
Node 3
Split 1 Address: Node 1
Split 2 Address: Node 1
Split 3 Address: Node 1
Split 4 Address: Node 1
Split 5 Address: Node 1
Split 6 Address: Node 1
Split 10 Address: Node 2
Split 11 Address: Node 3
Split 12 Address: Node 3
Split 7 Address: Node 1
Split 8 Address: Node 1
Split 9 Address: Node 1
Stage 2 : Compute Based Assignment
Max Splits Per Node: 6
Node 1
Node 2
Node 3
Split 1 Address: Node 1
Split 2 Address: Node 1
Split 3 Address: Node 1
Split 4 Address: Node 1
Split 5 Address: Node 1
Split 6 Address: Node 1
Split 10 Address: Node 2
Split 7 Address: Node 1
Split 8 Address: Node 1
Split 11 Address: Node 3
Split 12 Address: Node 3
Split 9 Address: Node 1
Stage 3 : Redistribution for Uniform Workload
Max Splits Per Node: 6
Node 1
Node 2
Node 3
Split 1 Address: Node 1
Split 2 Address: Node 1
Split 3 Address: Node 1
Split 4 Address: Node 1
Split 10 Address: Node 2
Split 7 Address: Node 1
Split 8 Address: Node 1
Split 5 Address: Node 1
Split 11 Address: Node 3
Split 12 Address: Node 3
Split 9 Address: Node 1
Split 6 Address: Node 1

Locality Aware Scheduling: Benefits

- Massive increase in the number of cached/local reads (around 9x)
- Reduced network traffic
- Improved Query perf


Performance comparison i3.4xLarge 10 nodes parquet1000
Presto Without RubiX : Presto-RubiX With OLS
Presto-RubiX With OLS
Presto-RubiX Without OLS
Presto Without RubiX
Execution Time (in seconds)
Queries
1.35
1.4
1.32
1.34
1.34
1.31
1.16
1.16
1.17
1.27
1.22
1.13
1.28
1.29
1.21
1.15
1.17
q03
q07
q18
q26
q27
q33
q34
q36
q43
q44
q46
q48
q55
q56
q68
q73
q79


Cache Reads comparison i3.4xLarge 10 nodes parquet1000
With OLS
Without OLS
Local Cache Reads Per Node
Queries
0
2000
4000
6000
8000
10000
12000
0
2
4
6
8
10
9.13
9.11
9.26
8.73
9.57
7.72
8.78
9.52
9.18
5.33
9.14
9.14
8.64
7.74
9.1
8.92
8.89
q03
q07
q18
q26
q27
q33
q34
q36
q43
q44
q46
q48
q55
q56
q68
q73
q79

Locality Aware Scheduling

- Has been adopted as the default scheduling model in Presto (release 315)
- Additional scheduler complexity: in the order of milliseconds

Thank you